



Beta Presentation

Vulnerability Scan and Detect

The Capstone Experience

Team McKesson

John Bannon

Brady Johnson

Chris Nguyen

Ananya Chittineni

Nicholas Felarca

Demetrius Wilson

Department of Computer Science and Engineering

Michigan State University



*From Students...
...to Professionals*

Spring 2025

Project Overview

- Website vulnerability scanner that detects web application security risks
- Scan, record, and exports vulnerability information to a database
- Capture and visualize vulnerability information
- Internal tool used to validate and maintain existing company web applications



Team Member's Technical Tasks

Technical Tasks Assigned

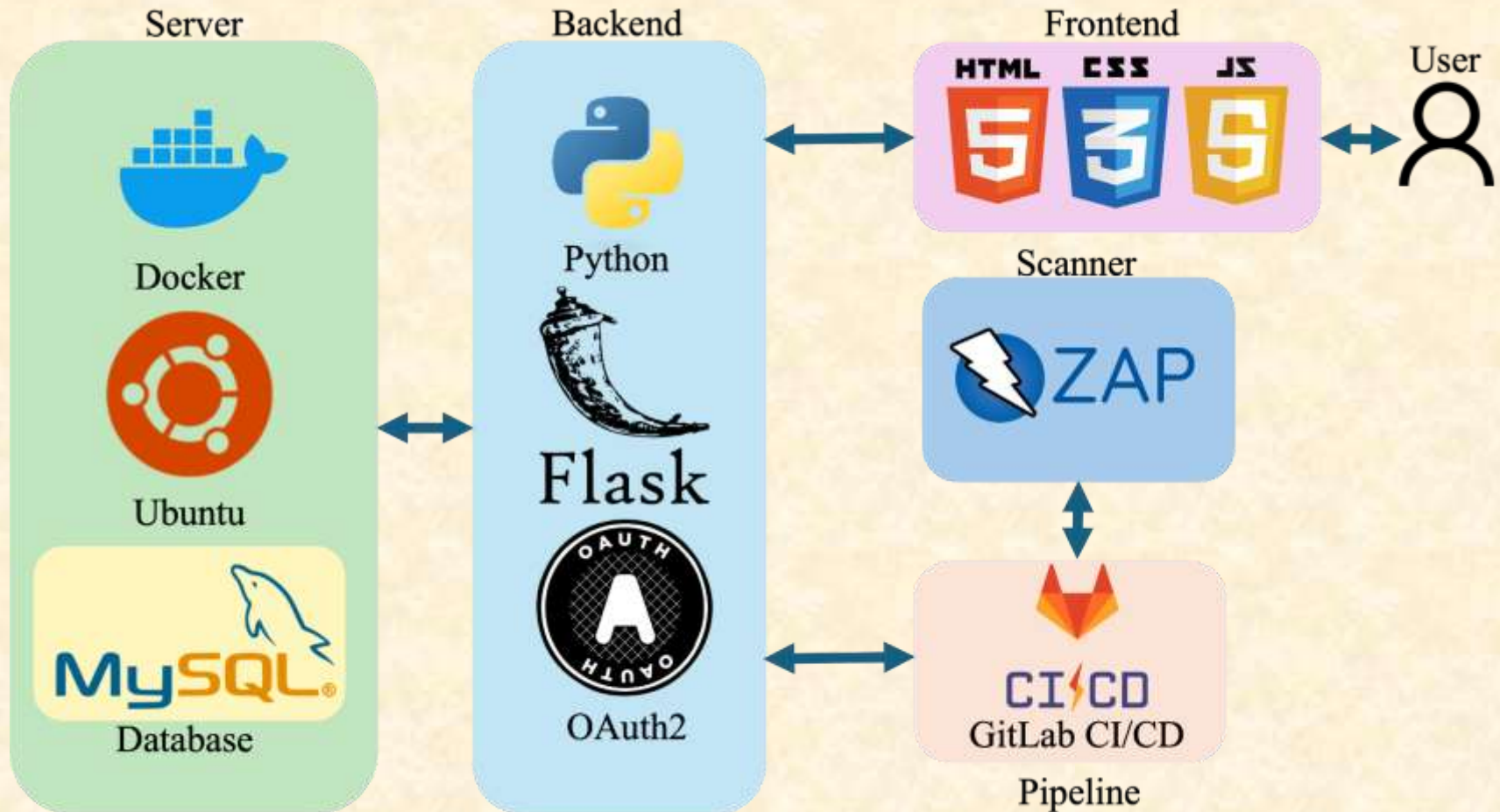
- John Bannon
 - Setup web app template
 - Setup MySQL database and connected it to Flask backend
 - Implemented layered authorization, share report with user feature, authorized domain management,
 - Implement feature where an admin can delete a website group and all of its associate scans
- Brady Johnson
 - Initial implementation for dashboard, new scan, and solutions pages
 - Frontend for advanced dashboard filters
 - Database queries to populate report
 - Dynamically update the report page with scan data
- Chris Nguyen
 - Integrate ZAP scanning engine with CI/CD pipelines and setup Gitlab runners to pick up jobs
 - Setup database schema and managed the database
 - Backend to get the dashboard dynamically updating.
 - Setup and worked on the overview, vulnerabilities, and websites tab on the reports page
- Ananya Chittineni
 - Frontend architecture and UI design
 - Initial frontend for the home, about, and results page
 - Theme support implementation for light mode and dark mode
 - Database query integration and UI design for trends tab in the reports page
- Nicholas Felarca
 - Download and parse raw scan data
 - Created website groups for URLs
 - Implemented a scheduler for users to schedule scans
 - Loading in website/dashboard data
- Demetrius Wilson
 - Connected CI/CD pipeline with the backend when a scan is initiated
 - OAuth2 login feature using google accounts
 - Search feature for dashboard page for website groups
 - Backend for advanced dashboard filter
 - Dynamically update the website tab on the reports page

Technical Tasks Completed

- John Bannon
 - Setup web app template
 - Setup MySQL database and connected it to Flask backend
 - Implemented layered authorization, share report with user feature, authorized domain management,
 - Implement feature where an admin can delete a website group and all of its associate scans
- Brady Johnson
 - Initial implementation for dashboard, new scan, and solutions pages
 - Frontend for advanced dashboard filters
 - Database queries to populate report
 - Dynamically update the report page with scan data
- Chris Nguyen
 - Integrate ZAP scanning engine with CI/CD pipelines and setup Gitlab runners to pick up jobs
 - Setup database schema and managed the database
 - Backend to get the dashboard dynamically updating.
 - Setup and worked on the overview, vulnerabilities, and websites tab on the reports page
- Ananya Chittineni
 - Frontend architecture and UI design
 - Initial frontend for the home, about, and results page
 - Theme support implementation for light mode and dark mode
 - Database query integration and UI design for trends tab in the reports page
- Nicholas Felarca
 - Download and parse raw scan data
 - Created website groups for URLs
 - Implemented a scheduler for users to schedule scans
 - Loading in website/dashboard data
- Demetrius Wilson
 - Connected CI/CD pipeline with the backend when a scan is initiated
 - OAuth2 login feature using google accounts
 - Search feature for dashboard page for website groups
 - Backend for advanced dashboard filter
 - Dynamically update the website tab on the reports page



System Architecture



Dashboard Page

The screenshot displays the McKesson Scan Dashboard. At the top, there is a navigation bar with links: Home, Dashboard, Solutions, About, Settings, and Logout. The user is logged in as `bannonj2@msu.edu`. The main section is titled "My Websites" and includes a table of scanned websites. The table has columns for Website Name, Website URL, Date, Status, and risk levels (High, Medium, Low, Informational). Each row has a "View Report" button. Below the table, there are four summary cards for risk levels: High Risk Issues (0), Medium Risk Issues (30), Low Risk Issues (116), and Informational Risk Issues (55). A final card shows "Total Websites Scanned: 4" with a breakdown: Successful: 2, Failed: 0, Running: 1.

Website Name	Website URL	Date	Status	High Risks	Medium Risks	Low Risks	Informational Risks	Actions
John final exam	https://exam-60184436772.us-central1.run.app	No scans	N/A	-	-	-	-	View Report
Juice Shop	https://juice-shop.hackuapp.com/	31 Mar 2025 16:50	running	0	0	0	0	View Report
Broken Crystals	https://brokencrystals.com/	28 Mar 2025 20:20	success	0	8	67	31	View Report
Brady's Website	https://homework-603200124830.us-central1.run.app/home	26 Mar 2025 16:21	success	0	22	49	24	View Report

High Risk Issues
0

Medium Risk Issues
30

Low Risk Issues
116

Informational Risk Issues
55

Total Websites Scanned: 4
Successful: 2
Failed: 0
Running: 1



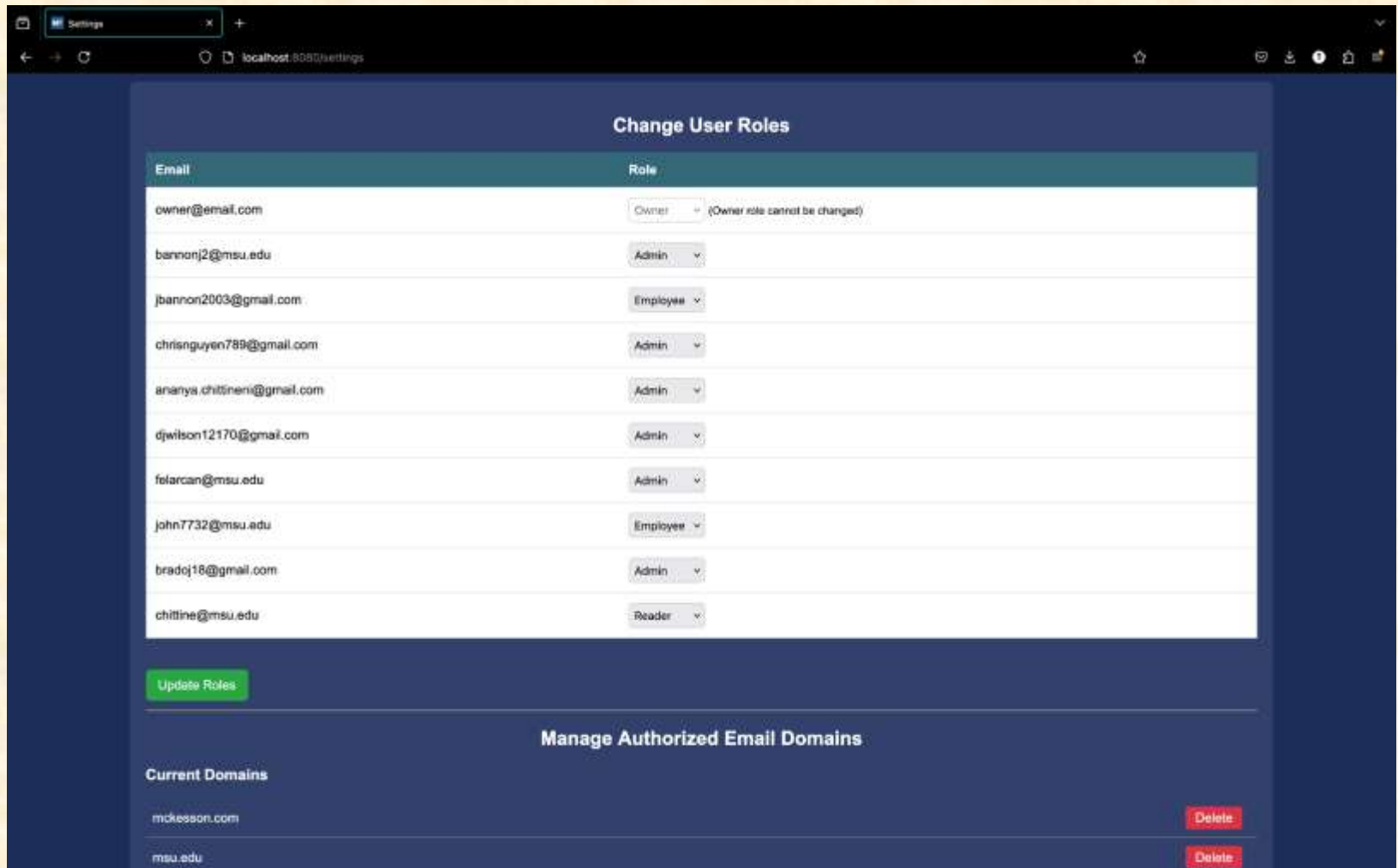
Website Group Page

The screenshot shows a web application interface for McKesson. The top navigation bar includes links for Home, Dashboard, Solutions, About, Settings, and Logout. The user is logged in as `bannonj2@msu.edu`. The main section is titled "Brady's Website" and contains a table of scan results. Above the table are buttons for "Quick scan", "Cancel", "Schedule Scan", "Manage Users", and "Filter Results".

Website URL	Date	Status	High Risks	Medium Risks	Low Risks	Informational Risks	Actions
<code>https://homework-600200124830.us-central1.run.app/home</code>	26 Mar 2025 16:21	SUCCESS	0	22	49	24	View Report Download JSON
<code>https://homework-600200124830.us-central1.run.app/home</code>	25 Mar 2025 10:38	SUCCESS	0	22	44	23	View Report Download JSON
<code>https://homework-600200124830.us-central1.run.app/home</code>	25 Mar 2025 00:28	SUCCESS	0	21	48	24	View Report Download JSON
<code>https://homework-600200124830.us-central1.run.app/home</code>	19 Mar 2025 17:39	SUCCESS	0	20	46	24	View Report Download JSON



Settings Page



The screenshot displays a web application interface for managing user roles and email domains. The browser window shows the address bar with 'localhost:8080/settings' and a tab titled 'Settings'.

Change User Roles

Email	Role
owner@email.com	Owner (Owner role cannot be changed)
bannonj2@msu.edu	Admin
jbannon2003@gmail.com	Employee
chrisnguyen789@gmail.com	Admin
ananya.chittineni@gmail.com	Admin
djwilson12170@gmail.com	Admin
felarcan@msu.edu	Admin
john7732@msu.edu	Employee
bradoj18@gmail.com	Admin
chittine@msu.edu	Reader

[Update Roles](#)

Manage Authorized Email Domains

Current Domains

mcKesson.com	Delete
msu.edu	Delete



Report Page



What's left to do?

- Stretch Goals
 - Headless API
- Other Tasks
 - Performance optimization
 - UI tweaks
 - Bug fixes

Questions?

?

?

?

?

?

?

?

?

?

